REMARKS

In an Office Action dated December 27, 2005, the Examiner objected to the specification; rejected claims 24 and 32 under 35 U.S.C. 112, second paragraph, as indefinite; rejected claims 24, 25, 29, 30 and 32 under 35 U.S.C. 103(a) as unpatentable over Doing et al. (US 5,161,166) in view of Kedem et al. (US 6,134,643) and Schumann et al. (US 6,012,106); and rejected claim 31 under 35 U.S.C. 103(a) as unpatentable over *Doing*, *Kedem* and *Schumann*, further in view of Patterson & Hennessy, "Computer Architecture: A Quantitative Approach" and Chauvel (US 6,957,315 B2). Claims 26-28, 33, and 34 were objected to as being dependent upon a rejected base claim, but otherwise indicated to contain allowable subject matter. Claims 1-23 were allowed.

A minor typographical error was corrected in claims 4 and 20.

Objections to the Specification

Two paragraphs of the specification have been amended to include the serial number of a related application. The reference to the assignee's docket number has been deleted as superfluous.

Indefiniteness

In response to the Examiner's indefiniteness rejection, applicant believes that the claims are sufficiently definite as written. Applicant's use of "a plurality" was intentional, and was meant to include either the previously referenced "plurality" or a subset thereof. Specifically, "said page table entries" is abundantly clear: there is one and only one previous reference to page table entries, these being the page table entries of the plurality of page table entries contained in the page table. However, in the second reference to "a plurality", this could refer to all of the

previously referenced plurality of page table entries contained in the page table, or to only some of them. I.e., the claims are sufficiently broad to cover any subset of the previously referenced "plurality of page table entries", so long as the subset contains multiple page table entries of the previously referenced "plurality of page table entries.[1] However, since the Examiner apparently believes the existing claim language is confusing, applicant has amended the claims to replace "a plurality" with "at least some".

Prior Art

Claim 32 has been cancelled, and the rejection thereof is moot. Claims 33 and 34 have been rewritten in independent form, incorporating all the limitations of previous claim 32, from which they depended. As amended, claims 33 and 34 are identical in scope to original claims 33 and 34, although now in independent form. Since these claims were indicated to contain allowable subject matter, and were objected to solely for being dependent on a rejected base claim, claims 33 and 34 as amended are allowable.

Applicant has amended independent claim 24 herein to clarify the scope of the present invention, if such clarification be necessary. In particular, the claim has been amended to clarify that the persistent reference history data is with respect to memory references generated by the at least one processor to memory locations in the corresponding addressable page of memory. As amended the claim is patentable over the cited art.

---

[1] Applicant is not trying to be unduly obscure, but believes that this construction is necessary to provide the proper degree of protection to his invention. Depending on various architectural considerations, some of the page table entries may contain initialized data, blank data, or other data. For example, in at least some page table designs, page table entries are sparse; at any instant in time, most of the page table entries are blank, indicating that there is no page in memory corresponding to a particular address. The claims as written require only that *at least some* of the page table entries contain the recited persistent reference history data.

Applicant's claim 24 recites one aspect of the invention disclosed herein. The key inventive feature recited in claim 24 is the ***persistent nature of the reference history data*** used for pre-fetching. Specifically, in accordance with applicant's claimed invention, reference history data persists for the duration of the time that a page is in main memory. The use of such persistent reference history data with respect to memory references by the processor or processors into a memory page is neither taught nor suggested by any of the cited art references.

Applicant's amended claim 24 recites:

> 24. A digital data processing device, comprising:
> at least one processor which generates memory references to memory address locations;
> a memory, said memory containing a page table, said page table having a plurality of page table entries corresponding to addressable pages, wherein each of at least some of said page table entries contains *persistent reference history data with respect to memory references generated by said at least one processor to memory address locations within the corresponding addressable page, said persistent reference history data being maintained throughout the life of the corresponding addressable page in memory*;
> at least one cache for temporarily storing data from said memory for use in satisfying said memory references generated by said at least one processor; and
> a pre-fetch engine, said pre-fetch engine pre-fetching data from said addressable pages to said at least one cache, said pre-fetch engine selecting data for pre-fetching using said persistent reference history data. [emphasis added]

As is known, data from a memory page is brought into cache on demand and evicted from cache when no longer needed. Various algorithms exist for tracking a cache line while in cache and determining when to evict it from the cache to make room for other data. Generally, these techniques require that the cache maintain some form of metadata with respect to the data held in cache. For example, in a well known least-recently-used (LRU) algorithm, the cache will maintain, with respect to each cache line, one or more bits of metadata indicating the recency of reference to the cache line. When a line must be selected for eviction from the cache, these bits of

metadata are used to select an appropriate line. Common to all these schemes is the fact that, once the metadata data is evicted from the cache, it is lost.

Kedem discloses a pre-fetch engine within the cache which contains a "prediction table cache". The prediction table cache is essentially a cache for some relatively small number of memory pages, and contains, for each page entry in the prediction table cache, the most recent N cache lines which were referenced.

Applicant does not contest that the data in *Kedem*'s "prediction table cache" is a form of reference history data. Leaving aside the fact that *Kedem*'s data is not contained in the page table as recited in claim 24, it is more significant that *Kedem*'s data is ***not persistent***. Specifically, *Kedem*'s prediction table cache entries are not "maintained throughout the life of the corresponding addressable page in memory", as recited in applicant's claim 24. As disclosed in *Kedem*:

> If the prediction table cache 40 were sized to hold an entry for each page in the DRAM 20, the array would be prohibitively large. Therefore, the prediction table cache 40 is sized depending on the execution engine 15 architecture to hold PTC entries 50 for a collection [i.e. a subset] of currently active pages. *Replacement algorithm for deciding which pages to track are known to one of ordinary skill in the art.* [Col. 3, lines 43-50, emphasis added]

From the above passage it is clear that *Kedem*'s prediction table cache holds prediction data for a selective subset of pages on a temporary basis, i.e., just like a cache. When a prediction table cache entry is evicted from the prediction table cache, the data is lost. Therefore *Kedem* does not disclose or suggest ***persistent reference history data*** as that term is defined and limited in claim 24.

Nor is the use of persistent reference history data taught or suggest by *Schumann*. *Schumann* discloses a system which maintains historical data concerning the size of direct

memory access (DMA) data transfers between memory and an I/O device. *Schumann*'s historical data is maintained in a DMA controller within a host bridge, and is apparently persistent while the applicable page remains in memory. However, this data is not used for pre-fetching cache lines, and does not reflect historical memory references from a processor. In other words, *Schumann*'s data is indeed persistent, but it lacks the other recited characteristics persistent reference history data, i.e., it does not show the history of "memory references generated by said at least one processor to memory address locations within the corresponding addressable page", as recited in amended claim 24.

So what do these references show together? One shows data of a type recited by applicant, but which is not persistent. The second shows data of a completely different type, which is persistent. Applicant submits that the references taken together can not possibly suggest that data of the first type be made persistent. Persistence of data is intimately connected with the type of data, how it is maintained, where it is maintained, and so forth. It is well known that some forms of data persist for a long time. Just because some unrelated data happens to be persistent does not necessarily suggest that applicant's recited "reference history data with respect to memory references generated by said at least one processor..." can or should be persistent.

Far from suggesting the use of persistent reference history data, *Kedem* teaches away from persistence. As the above quoted passage illustrates, *Kedem* teaches that it is impractical to hold all the required history data in the prediction table cache, and that it is necessary to maintain history data for only a relatively small subset of pages, evicting these page entries according to some appropriate replacement algorithm. Eviction of page entries necessarily results in loss of the data, and persistence as recited by applicant is lost.

The third reference, *Doing*, is cited to show various basic architectural features of cache memory and memory accessing structures. The Examiner apparently concedes that it does not disclose a pre-fetch engine or the use of persistent reference history data.

For the above reasons, claim 24 as amended is not obvious over *Kedem*, *Schumann* and *Doing*. Claims 25-31 are dependent on claim 24 and patentable for the same reasons.

In view of the foregoing, applicant submits that the claims are now in condition for allowance, and respectfully requests reconsideration and allowance of all claims. In addition, the Examiner is encouraged to contact applicant's attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

DAVID A. LUICK

By:_____
Roy W. Truelson
Registration No. 34,265

Telephone: (507) 202-8725